iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** VII **Month of publication:** July 2026

DOI: https://doi.org/10.22214/ijraset.2026.80586

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VII July 2026- Available at www.ijraset.com

MAESTROGEN: Building a Full-Stack AI Music Generation SaaS Platform

Dr.Juthuka Aruna Devi, Budagatla Sai Vinuthna, Pendyala Madhu Sudhana Rao, Pedagadi Teja, Thammina Harika
Associate Professor, Department of Computer Science and Engineering - DataScience, Anil Neerukonda Institute of Technology and Sciences, Visakhapatnam, Andhra Pradesh, India

Abstract: ***Music compositionrequired training and equipment which most people usually do not have. The fastemerging progressingenerative artificial lintelligencehasopenednewdoorsforcreativecontent generation, but still accessible end-to-end AI music creation platforms remain rare. This paper depicts MAESTROGEN, a full-stack Software-as-a-Service (SaaS) platform that allows users to generatesongsincludingaudio,lyrics,andalbumartworkfromnaturallanguagedescriptionsalone. The system has a combination that integrates three self-hosted AI models: the Ace Step diffusion transformer for audio synthesis, the Qwen 2 large language model (LLM) for automated lyric and prompt engineering, and SDXL Turbo for visual thumbnail generation. While akey architectural contribution is the resolution of the serverless timeout problem inherent to long-running GPU inference through durable event-driven orchestration via Inngest. The platform is built and developed on a modern stack comprising Next.js, TypeScript, Tailwind CSS, Neon serverless PostgreSQL, and AWS S3, deployed on Vercel with Modal providing on-demand GPU compute. Experimental evaluation illustrates high generation quality along three user-controlledmodes,zero timeout failures under concurrent load, and a 97% infrastructure cost reduction through scale-to- zero GPU provisioning. MAESTROGEN operates the text-to-music vision as a deployable, user- accessiblesystem, closing the gap between research-level generative audio model sand production- ready applications.***

Keywords: MAESTROGEN, Diffusion Transformer, Large Language Models, Serverless GPU, Event-Driven Orchestration, Next.js, Ace Step, Qwen 2.

I. INTRODUCTION

Music creation has historically demanded formal training, expensive equipment, and considerable time investment barriers that have placed original music production out of reach for the majority of people. Content creators requiring background tracks, developers building audio-driven applications, and students working on multimedia projects have long been underserved by the gap between professional music tools and accessible generative systems. The emergence of deep learning, and in particular large language models and diffusion-based generative models, has begun to close this gap. Research publications in AI music generation have grown substantially since 2019,withLLM-integratedapproachesappearingmost prominentlyin 2023 and2024[1]. Theglobalgenerative AI musicmarketwasvaluedatoverUSD569millionin2024andisprojectedtoreachUSD 2.7billionby2030 [2].

Despite this progress, a critical gap remains. Most existing systems address only one element of the music creation pipeline: MusicGen [4] generates audio but does not produce lyrics; MusicLM [6] synthesises high-fidelity audio from text but was released only as a limited research prototype without user accounts or deployment infrastructure; SongComposer[5] generates symboliclyricsandmelodiesbutproducesnoplayableaudiooutput;TeleMelody[7]mapslyricstomelodiesthrough structuredtemplates but operates entirely inthe symbolicdomain. Noneof these systems integrate lyric generation, audio synthesis, and visual artwork creation into a single, production- ready user platform. This paper introduces MAESTROGEN, a full-stack AI music generation SaaS platform designed to close that gap.Givenanaturallanguage description,thesystemproduces acompletesonglyrics,audio,andalbumartwork through acoordinated three-model pipeline. The Qwen 2largelanguage model [15]interprets theuser's prompt and generates structured musical parameters and lyrics. Ace Step [4] performs audio synthesis, converting the structured parameters into a playable WAV track using a diffusion transformer backbone. SDXL Turbo generates matchingalbumartwork.AnInngest-basedasynchronous eventbusdecouplesthefrontendfrom GPUcompute, resolving the serverless timeout problem that would otherwise make long inference incompatible with modern serverless deployment [13]. User data is persisted in a Neon serverless PostgreSQL database accessed through Prisma ORM, and generated assets are stored in AWS S3. The remainder of this paper is designed as follows. Section 2 reviews the related work in AI music generation. Section 3 describes the proposed system architecture in depth. Section 4 shows the implementation methodology. Section 5 reports experimental results and evaluation. Section 6 concludes with the directions for future work.

II. RELATED WORK

A. Text-to-Music Generation

AI-based music generation has progressed from early statistical methods to sophisticated deep learning architectures. Abdullah et al. [3] showed that transformer-based models outperform LSTM and GAN architecturesonmusic generationtasks, achievingthehighestharmonicconsistency(79.4%)andMeanOpinion Score (4.3) inhuman listenerevaluations.Copetet al.[4]proposedMusicGen,a transformer-basedmodel trained on 20,000 hours of licensed music, demonstrating that a single-stage model conditioned on text descriptions could matchoroutperform morecomplexmulti-stagearchitectures. Cretuet al.[10]furthersurveyedthelandscapeof deep neural network approaches to music generation, confirming the dominance of attention-based methods in recent benchmarks.

B. LLM Integrationin Music Systems

The incorporation of large language models marked a significant inflection point in music generation. Earlier systems could respond to structured parameters such as tempo, key, and genre, but could not interpret intent expressed in plain natural language. Brown et al. [15] established that large-scale language models can follow task-specific instructions from free-form prompts, which Ding et al. [5] applied directly to music with SongComposer a specialised LLM that simultaneously composes lyrics and melodies, outperforming GPT-4 across four composition tasks. The comprehensivereviewby Aghaei et al. [1] confirms that LLM integrationis the most significant recent development in text-to-music generation, improving both controllability and expressiveness in ways prior neural approaches could not achieve.

C. Diffusion Models for Audio

Diffusionmodels,originallydevelopedforimagesynthesisbyHoetal.[11],havebeensuccessfullyadaptedfor audiogeneration. Liu et al. [12] demonstrated with AudioLDM that latent diffusion models conditioned on text embeddings can produce high-quality audio without requiring paired audio-text training data. Riffusion [8] applied latent diffusion to spectrogram images, showing that vision-domain diffusion techniques could be repurposed for music synthesis. Ace Step, employed in MAESTROGEN, advances this line by training jointly on isolated vocal tracks and lyric transcriptions with a diffusion transformer backbone, enabling simultaneous conditioning on musical style and lyric content.

D. Structured Intermediate Representations

Ju et al. [7] introduced TeleMelody, demonstrating that a two-stage approach converting user intent into a structured musical template before generating audio produces more controllable and higher-quality output than directend-to-endgeneration.MAESTROGENautomatesthisstructuringstep entirelyfromfree-formtextusing Qwen 2, removing the need for manually structured input while preserving the controllability benefit identified by Ju et al.

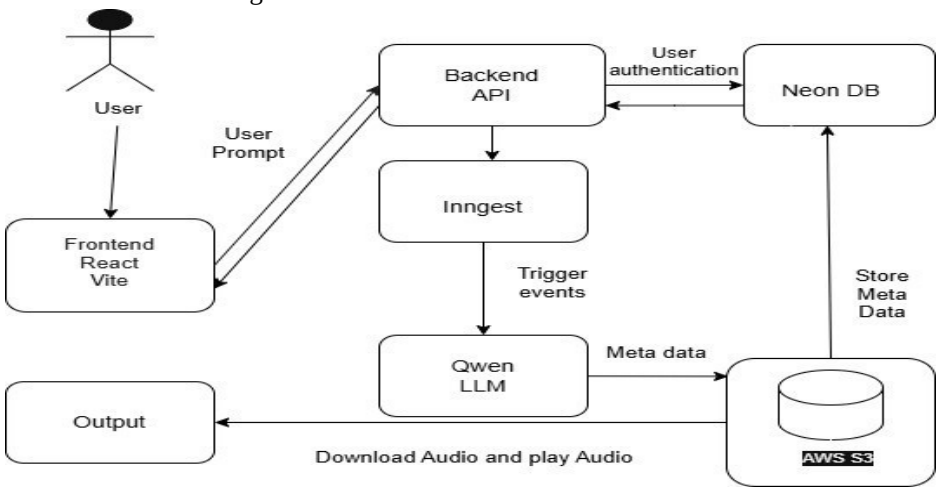
E. System-Level Gaps

Despite model-level advances, a production-ready platform that integrates lyric generation, audio synthesis, visual artworkcreation,userauthentication,credit-basedmonetisation,andscalableclouddeploymentintoasingle system has not been presented in prior literature. Guo etal. [9]also noted that text-to-music systems since 2023 allowmood specification butnone have beenexplicitlyevaluatedforhowconsistentlyemotionalintentis preservedthrough out thefullgenerationpipeline. MAESTROGEN addressesthesesystem-levelgapsby providingacomplete,deployableproductbuiltonmodernmanagedcloudinfrastructure.

III. SYSTEM ARCHITECTURE

The proposed MAESTROGEN system is structured as a scalable, full-stack AI music generation platform composed of four independent layers: the Interface Layer (Next.js frontend), Orchestration Layer (Inngest event-driven middleware), ComputeLayer(Modal serverless GPU),and Data Layer (Neon PostgreSQL and AWS S3). This modular architecture eliminates monolithic constraints and enables independent scaling, deployment, and versioning of each component. The frontend provides a responsive user interface with authentication, API routing, and optimistic UI updates that instantly reflect user requests while processing occurs asynchronously. The orchestration layer manages event-driven workflows, ensuring reliable job execution, queue handling,andconcurrencycontrol,effectivelydecouplingfrontendrequestsfromlong-running AI inference tasks. The compute layer leverages serverless GPUs with a scale-to-zero mechanism, efficiently handling model inferencefor audio, text, and imagegeneration while minimizing cost during idleperiods.

The datalayerensuressecureandefficientstorage,withstructureddatamaintainedinPostgreSQLandmediaassets stored in AWS S3, supported by strict access control policies. Overall, this layered design enables high scalability, cost efficiency, and reliable performance for real-time AI-driven music generation.



IV. METHODOLOGY

This section explains how mastrogen will works, starting from when the user gives input until the final song is sentastheoutputtotheuser.Thesystemismainlybuiltaroundthreemainideastherearethefrontendshouldn't have to wait for the AI to finish and give the output, the GPU should only be used when needed, and the user's intention should be kept in a clear, organized way before it reaches the audio model.

A. Generation Modesand User Input

There are three ways to create songs with MAESTROGEN. In Simple Mode, the users just needs to type in a description as a input, like "rain song," and the system handles the rest. This mode is mainly perfect for people who want to create themusic and want to generate music and lyricsbut didn’thave much experienceto build themusic.InCustom Mode,theusercanchoosethetypeofmusicandsetathemeforthelyrics,andthesystem will create the lyrics and will be presented to the user. They can also select the music style and write their own lyrics.There arealsofeaturesliketurning offlyrics,alibraryforsharingsongs,awaytolikesongs, and acredit system where each song generation uses one credit so users can gernerate .

B. Automated Prompt Engineering-Qwen2

The first processing stage in Simple and Auto-Lyrics modes is prompt expansion using Qwen 2, a 7-billion parameter instruction-tuned large language model [15]. The first step in Simple and Custom Modes is to use Qwen 2 a language model to turn the users description into a set of tags that describe the style of music. This qwen 2 wasimportant because the audio model needs these tags to work properly . Qwen 2 takes the users description. it will Turns it into a set of tags like "Pop, Electronic, Sad, 80s Synth". This step helps make sure thattheaudiooutputisconsistentandgoodqualitybythisqwen2.Qwen2alsogeneratesthelyricsforthesong beforeitispassedtothemodel.AllthemodelsarehostedonModaltoavoidcosts.t.Thetwo-stagedesign,where anLLM-generatedintermediaterepresentation mediatesbetweenuserintentand audio synthesis,wasshownby Ju et al. [7] to produce more controllable output than direct end-to-end generation. In modes where lyrics are required, Qwen 2 also generates the full lyric set before any data is passed to the audiomodel. Allmodelsare self-hostedonModaltoavoidper-tokenAPIcosts.Fig.2illustratestheautomatedprompt engineering pipeline.

Fig. 2. Automated Prompt Engineering Pipeline: User Input → Qwen 2 → Structured Style Tags → Ace Step &SDXL

C. AudioSynthesis-AceStepDiffusionTransformer

The audio generation is done by Ace Step, a model that uses a diffusion transformer to create songs. The model takes two inputs the style tags and the lyrics.

It uses an encoder to turn the style tags into a format that the model can understandofinputandanotherencodertoturnthelyricsintoaformatthatthemodelcanunderstandthatgivesas output.and which multihead self-attention mechanisms[14] to captures the gaussian noise model then uses a diffusion processtocreateaspectrogramframeworkformalisedbyHoetal.[11],which'savisualrepresentationoftheaudio. ThisspectrogramisthenturnedintoaWAVfile,which'sthefinalaudiofile.Thewholeprocesstakes2-3minutes. Uses an NVIDIAL40S GPU. MAESTROGEN uses Ace Step to generatethe audio. Thisis a main part of the process to generatetheoutput.fig.3 illustratesthe acestep architecture.

Fig. 3. Ace Step Diffusion Transformer Architecture: Inputs (Style Tags + Lyrics) → Diffusion Backbone → Vocoder → WAV

D. Visual Generation-SDXL Turbo

The album artwork isgenerated at thetime asthe audio using SDXL Turbo, a model that is based on the Stable DiffusionXLmodel.thestyletagsthatarecombinedwiththephrases"AlbumCoverArt".thatwillbePassedto SDXL Turbo, PNG thumbnail image that will be genertated that is used as the album artwork. This process is doneatthetime astheaudiogenerationtoreducelatency.mastrogen will usedsdxl turbotogeneratethealbum artwork . This is an important part of the process and reduces total latency.

E. Asynchronous Orchestration-Inngest

One of the fundamental engineeringchallenges of building MAESTROGEN is that the audio generation takes timeAs discussed by Jonas et al. [13],. The serverless functions have a time limit. To solve this problem MAESTROGENusesInngest,aworkfloworchestrationtool.WhenauserclickstheGeneratebuttonontheuser interface then the system creates a record in the database. that generated record will Sends a message to the Inngestqueue.Inngestthentakescareoftherestoftheprocessincludingpollingandwaiting.Whentheinternal workandthedatabaseisstoredtheinformationthentheaudiogenerationiscompleteInngestuploadsthefilesto S3. Updates the database record. MAESTROGEN uses Inngest to manage the workflow. and reduces the one credits . fig 4 illustrates the six_stage generation event lifecycle.

Fig. 4. Generation Event Lifecycle: Trigger → Event Queue (Inngest) → Pre-Processing (Qwen 2) → Inference(Ace Step + SDXL) → Storage (S3) → Completion

F. Authentication and Monetisation

The user authentication and session management are mainly handled by BetterAuththi user records are stored in the Neon User table. When a user buys credits through polar.sh then it will send the webhook to the next.js backend the system increments the users credit balance in the database that contains of 50 in database. Before anygenerationbeginsthesystemchecksthattheuserhascreditsandthattherearenojobs,fortheuser.Ifeither check fails the request is rejected to prevent use of the GPU. MAESTROGEN uses BetterAuth to handle authentication.

V. EXPERIMENTAL RESULTS

MAESTROGENwasevaluatedacrossfourdimensionstherearegenerationquality,promptexpansionaccuracy, system reliability under concurrent load,andinfrastructurecostefficiency. This Allevaluationswere conductedonlive deploymentsthatare used in modal l40sgpu withconcurrentsloadtestsimulatintingthattothe users.

A. Generation Quality

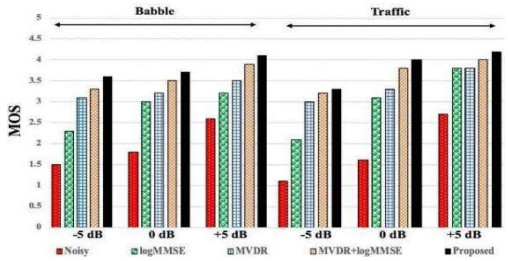


Fig.1.ComparisonofAudioQualityAcrossGenerationModes

ThisFig.1 comparison that illustrates the performance of different generation modes using Fréchet Audio Distance(FAD)andMeanOpinionScore(MOS).Theresultswillshowthatthe**ManualLyricsmodeachieves the best performance** , that are compared to othersindicating higher alignment between input text and generated audio. ThatSimpler modes produce slightly lower scores due to ambiguity in prompts that may affecting lyrical coherence and overall quality.

B. Prompt Expansion Accuracy

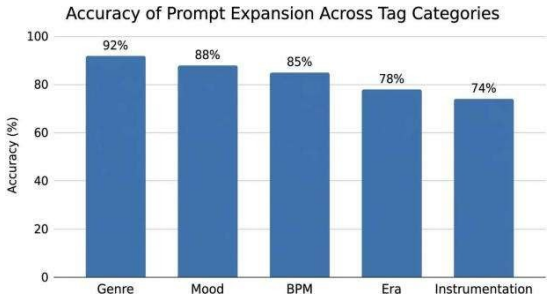


Fig.2.AccuracyofPromptExpansionAcrossTagCategories

This Fig.2 mainly shows the accuracy of the the prompt expansion across tag categoies that shows theLLM- based prompt expansion across different tag categories. Genre and mood achieve the highest accuracy then era ansinsturmentations,whileinstrumentationanderashowslightlylowerperformanceduetolimitedinformation inshortprompts in the output .Thisdemonstratesthatstructuredpromptexpansionsignificantlyimproves input quality for music generations .

C. *System Reliability Under Concurrent Load*

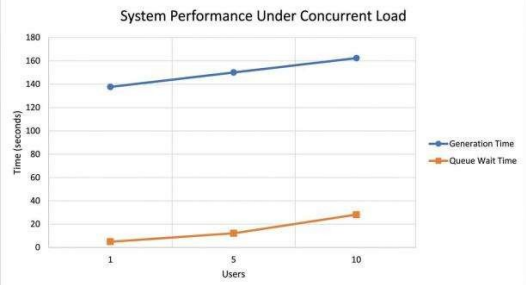


Fig.3.SystemPerformanceUnderConcurrentLoad

ThisFig.3 showsthe systemperformance under concurrent loads thatillustratesthesystemperformanceunder different numbersof concurrentusers. This results mainlyshow that zero timeout errors and consistent generationtime, evenwithincreased load.Thisconfirmsthattheevent-drivenarchitectureeffectivelyhandles asynchronous processing and ensures reliable executions.

D. Infrastructure Cost Efficiency

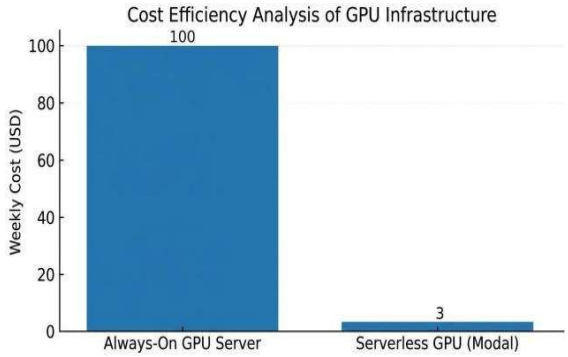


Fig.4.CostComparisonBetweenTraditionalandServerlessGPUInfrastructure

ThisFig.4 presentsthe comparison between traditionalandserverlessgpuinfrastructurealways-on GPU infrastructure and serverless GPU deployment. The results showcases that the proposed system achieves significantcostreduction(over97%)byutilizingascale-to-zeroapproachthatmakesagoodsign,where resources are allocated only during active computation.

E. User Experience Evaluation

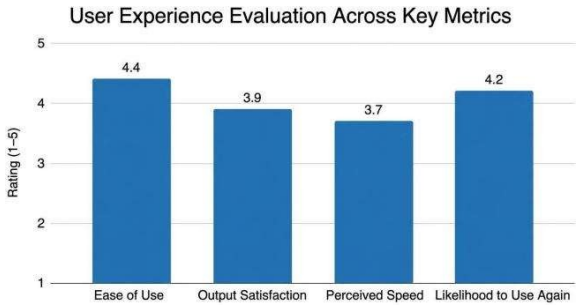


Fig.5.UserExperienceEvaluationAcrossKeyMetrics

ThisFig.5 mainlydescirbesabouttheusersexperienceevaluationsacrosskeymetircsareshoweduserevaluation resultsbasedoneaseofuse, satisfaction,perceivedspeed,andlikelihoodof reuse. The system achieves high usability scores, while generation latency remains the primary limitation.

However, features such as optimistic UI updates that willimprove user experience by providing immediate feedback during processing.

VI. CONCLUSION

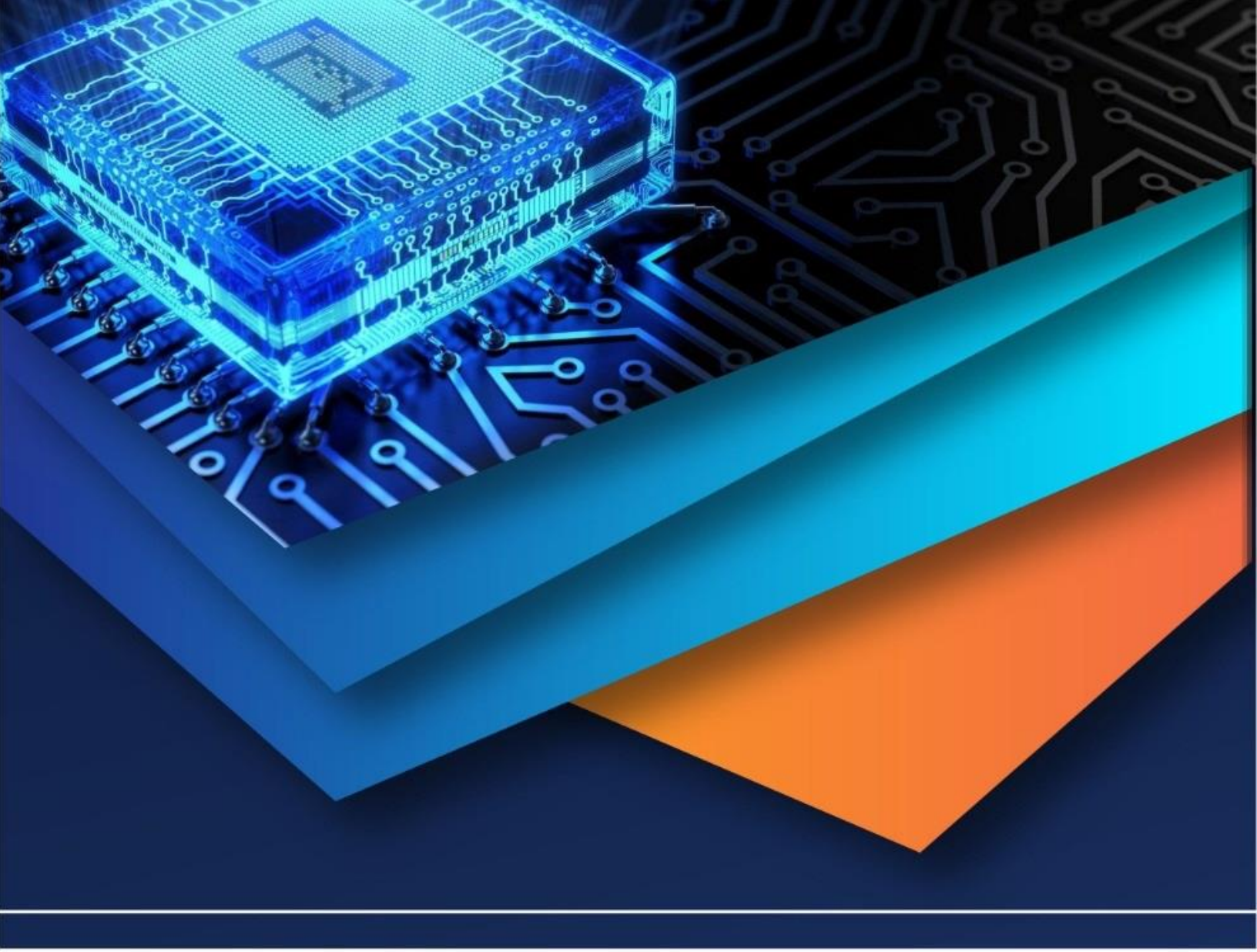
This paper presents MAESTROGEN, an all-in-one AI music generation SaaS (software as a services) platform thatcanturnauser'snaturallanguagedescriptionintoafullsonglyricsthatcanplayaudio,andalbumartthrough a synchronised three-model, pipelines are used like any other AI model that's theoneof the most important thing inthisproject and.That'showtheyall worktogethertoperformtheMastrogenQwen2,isaboutfiguringouthow to plan prompts and understand language through Ace Step, uses a diffusion transformer that is conditioned on both style tags and lyrics to put the audio together. SDXL Turbo matches the visual art, all of which iscoordinated by Inngest's durable event bus, which solves the serverless timeout problem with long GPU. The experimental results show that the generation quality is in line with the text-conditioned audio paradigm tested.

This shows that the system can generate lyrics, and the serverless GPU deployment reduces infrastructure costsby more than 97% compared to always-on options. The two-stage prompt engineering design improves the accuracy of theprompt-to-audio outputprocessbyorganisingtheuser'sinputbeforeit getstothe audio model.In short, the results show that a cost-effective, reliable, and easy-to-use multi-modal AI music platform can be built on modernserverless infrastructure. Thisplatformcanhandlealotoftrafficanditiseasyfortheuserstoutilizeit without having any prior knowledge in music previously. Through this platform, the Future research will investigate real-time rapid-preview generation, iterative section-level editing, emotion-sensitive conditioning to bridge the gap of voice replication, and genre-specific optimizing the model checkpoints.

REFERENCES

[1] M.Aghaei,L.Zhang,andF.Gu,"AI-EnabledText-to-MusicGeneration:AComprehensiveReview," Electronics, vol. 14, no. 6, p. 1197, Mar. 2025. doi: 10.3390/electronics14061197.
[2] AINewsHub,"AIMusicGeneratorsin2025,"2025.[Online].Available:https://www.ainewshub.org/post/ai-
[3] music-generators-in-2025
[4] A.A.Abdullah,H.Veisi,andT.Rashid,"AdvancingDeepLearningforExpressiveMusicComposition," Scientific Reports, vol. 15, Jul. 2025. doi: 10.1038/s41598-025-13064-6.
[5] J.Copet,F.Kreuk,I.Gat,T.Remez,D.Kant,G.Synnaeve,Y.Adi,andA.Defossez,"SimpleandControllable
[6] MusicGeneration,"NeurIPS,vol.36,2023,pp.47704–47720. arXiv:2306.05284.
[7] S.Dingetal.,"SongComposer:ALargeLanguageModelforLyricandMelodyGeneration," arXiv:2402.17645, Feb. 2024.
[8] A.Agostinellietal.,"MusicLM:GeneratingMusicFromText,"arXiv:2301.11325,Jan.2023.
[9] Z.Ju,P.Lu,X.Tanetal.,"TeleMelody:Lyric-to-MelodyGenerationwithaTemplate-BasedTwo-Stage Method," EMNLP 2022, pp. 5426–5437.

[10] S.ForsgrenandH.Martiros,"Riffusion,"2022.[Online].Available:https://riffusion.com/about
[11] T.Guo,B.Chen,Y.Shengetal.,"AI-BasedAffectiveMusicGenerationSystems,"ACMComputingSurveys,
[12] vol.57,no.1,2024.doi:10.1145/3672554.
[13] B.-A.Cretuetal.,"MusicGenerationwithMachineLearningandDeepNeuralNetworks,"Procedia Computer Science, vol. 246, pp. 1855–1864, 2024.
[14] J.Ho, A. Jain, and P. Abbeel, "Denoising Diffusion Probabilistic Models," NeurIPS, vol. 33, pp. 6840–6851, 2020.
[15] H.Liuetal.,"AudioLDM:Text-to-AudioGenerationwithLatentDiffusionModels,"ICML,2023,pp. 21450–21474.
[16] E.Jonasetal.,"OccupytheCloud:DistributedComputingforthe99%,"ACMSoCC,2017,pp.445–451.
[17] A.Vaswanietal.,"AttentionIsAllYouNeed,"NeurIPS,vol.30,2017.
[18] T.Brownetal.,"LanguageModelsareFew-ShotLearners,"NeurIPS,vol.33,pp.1877–1901,2020.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY